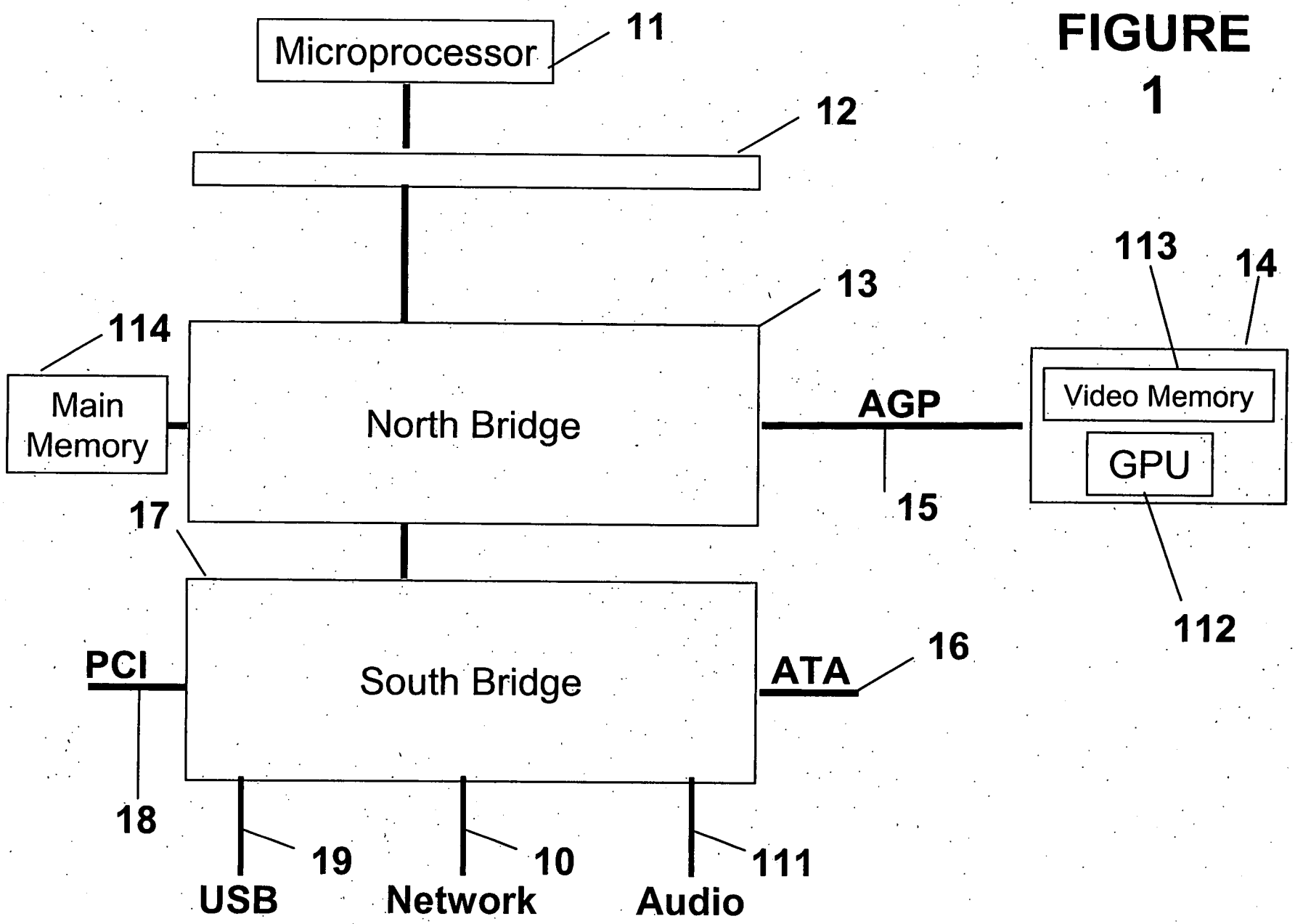

Microprocessor
11
12
113
14
13
114
Main Memory
North Bridge
AGP
Video Memory
GPU
15
17
112
PCI
South Bridge
ATA
16
18
19
10
111
USB
Network
Audio

FIGURE 1

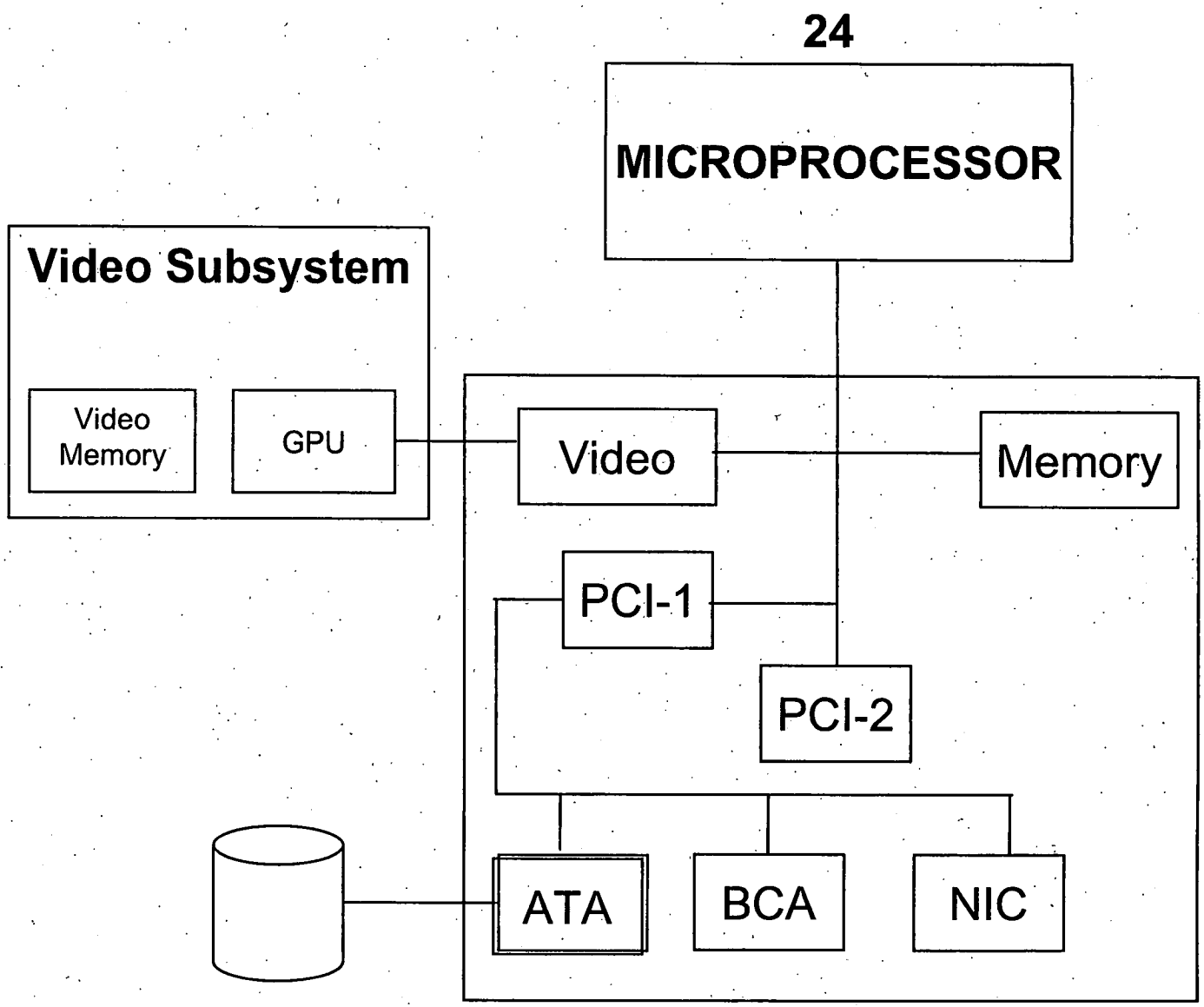

24
MICROPROCESSOR
Video Subsystem
Video Memory
GPU
Video
Memory
PCI-1
PCI-2
ATA
BCA
NIC

FIGURE 2(a)

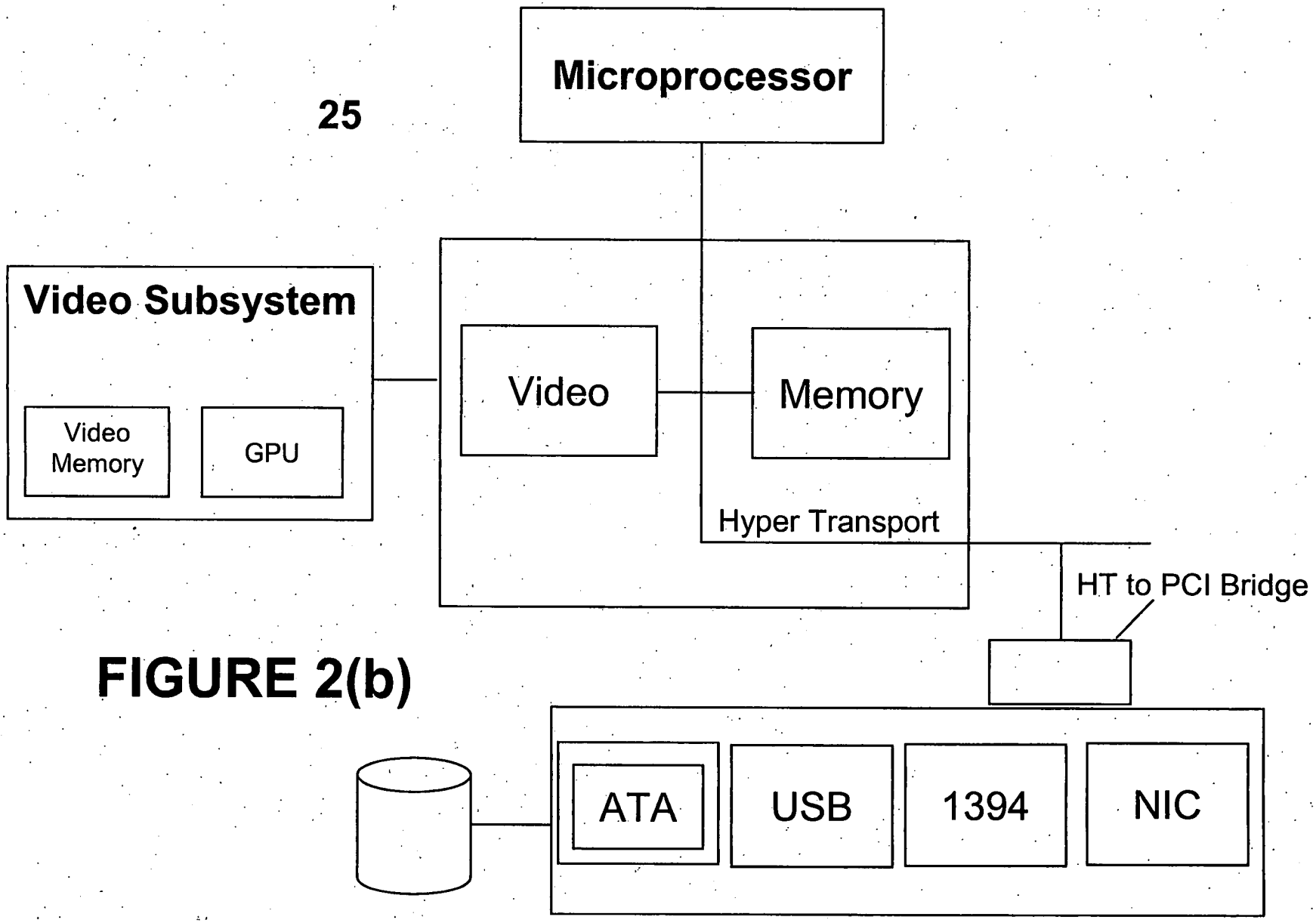

25
Microprocessor
Video Subsystem
Video Memory
GPU
Video
Memory
Hyper Transport
HT to PCI Bridge
ATA
USB
1394
NIC

FIGURE 2(b)

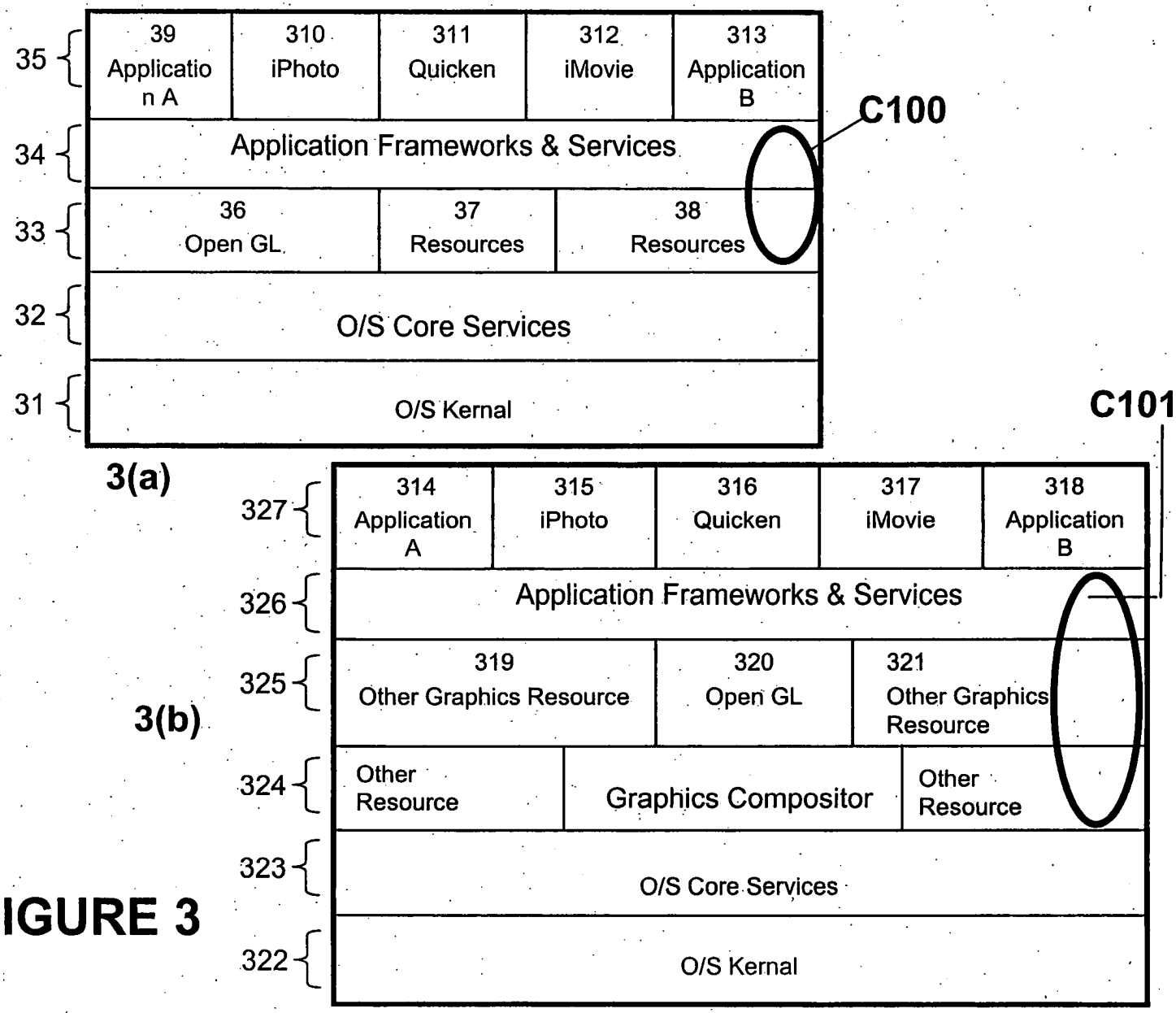

35
39 Applicatio n A
310 iPhoto
311 Quicken
312 iMovie
313 Application B
34
Application Frameworks & Services
33
36 Open GL
37 Resources
38 Resources
32
O/S Core Services
31
O/S Kernal
C100
3(a)
C101
327
314 Application A
315 iPhoto
316 Quicken
317 iMovie
318 Application B
326
Application Frameworks & Services
325
319 Other Graphics Resource
320 Open GL
321 Other Graphics Resource
3(b)
324
Other Resource
Graphics Compositor
Other Resource
323
O/S Core Services
322
O/S Kernal

FIGURE 3

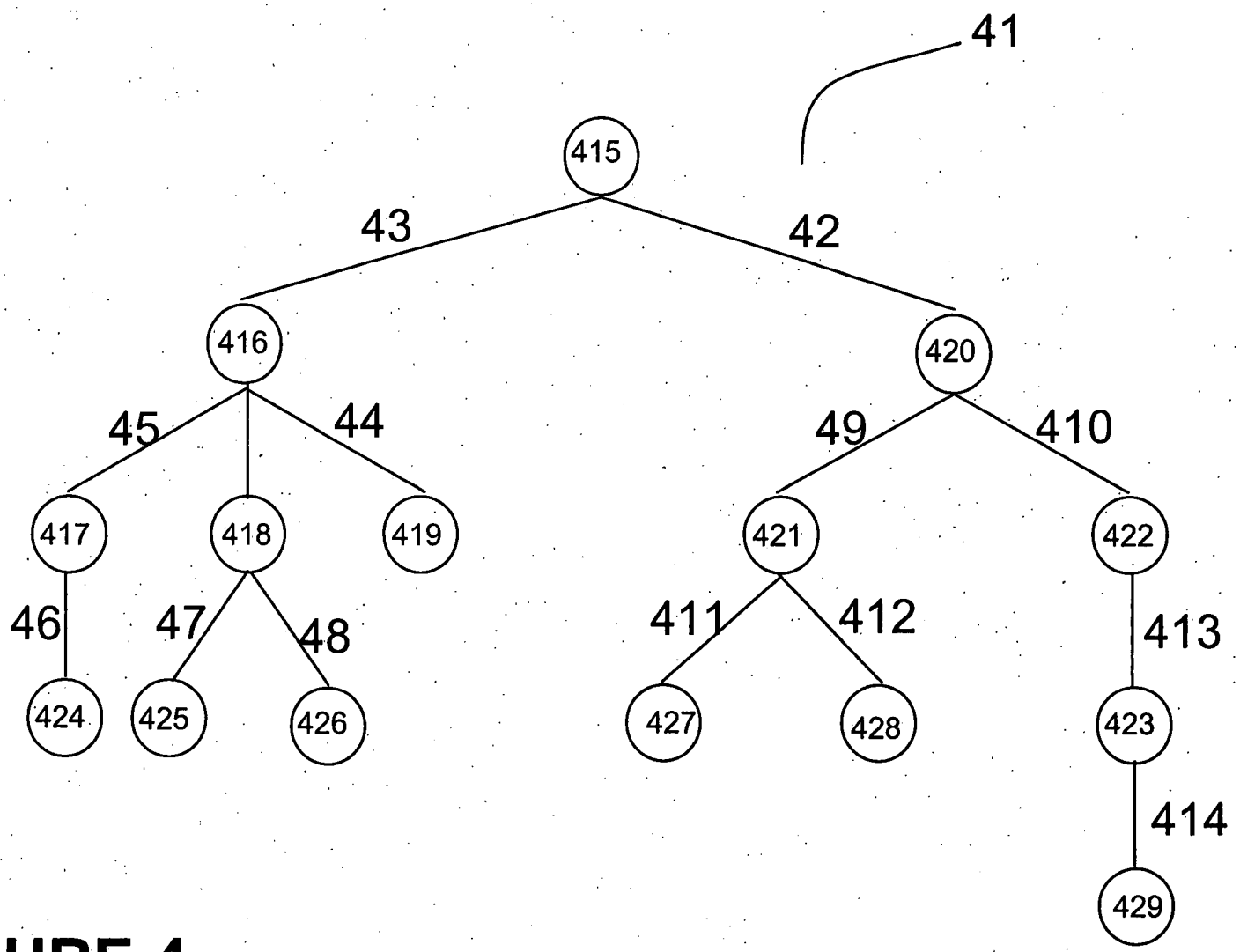
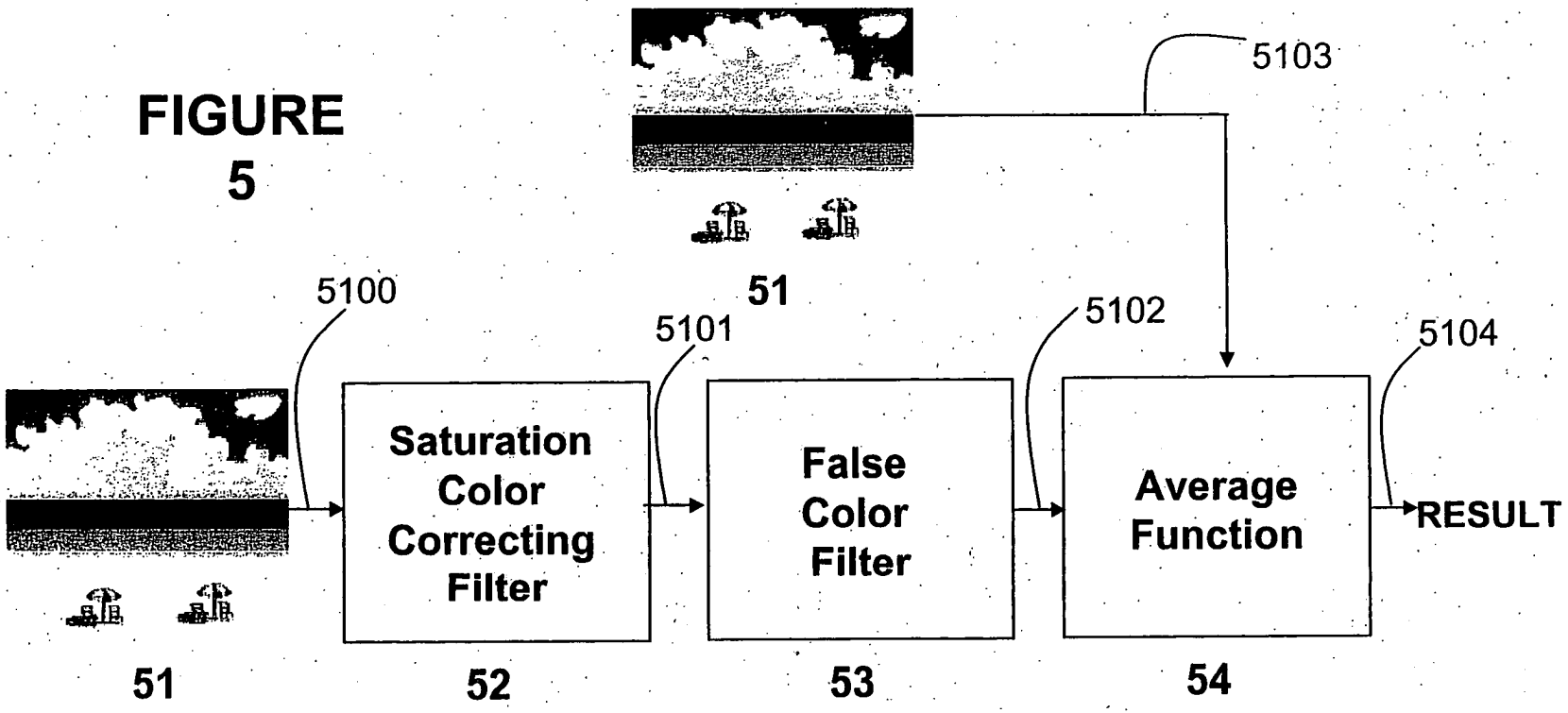

41
415
43
42
416
420
45
44
49
410
417
418
419
421
422
46
47
48
411
412
413
424
425
426
427
428
423
414
429

FIGURE 4

FIGURE 5



High-level Code Examples:

1. Allocate 51
2. Apply 52 to 51, parameters = (X, Y, Z, W), input = 51 (Sea Shore image), output = [place holder] CC sea shore
3. Apply 53, parameters (X, Y, Z, W), input = cc sea shore, output = [place holder] FC CC sea shore
4. Apply 54, parameters (X, Y, Z, W), input buffer 53, input Sea Shore image, output = [place holder] sea shore result

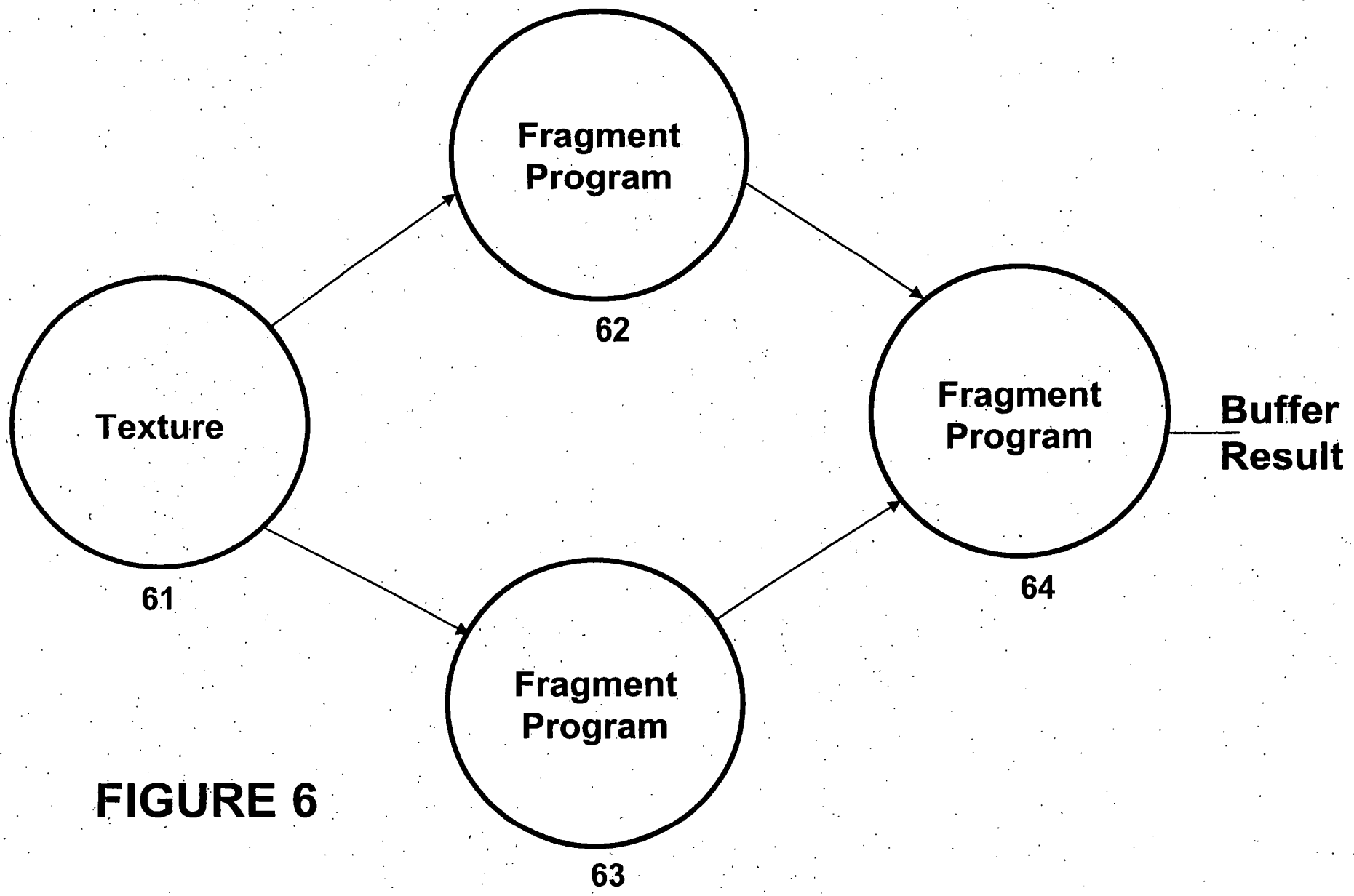

Fragment Program
62
Texture
61
Fragment Program
64
Buffer Result
Fragment Program
63

FIGURE 6

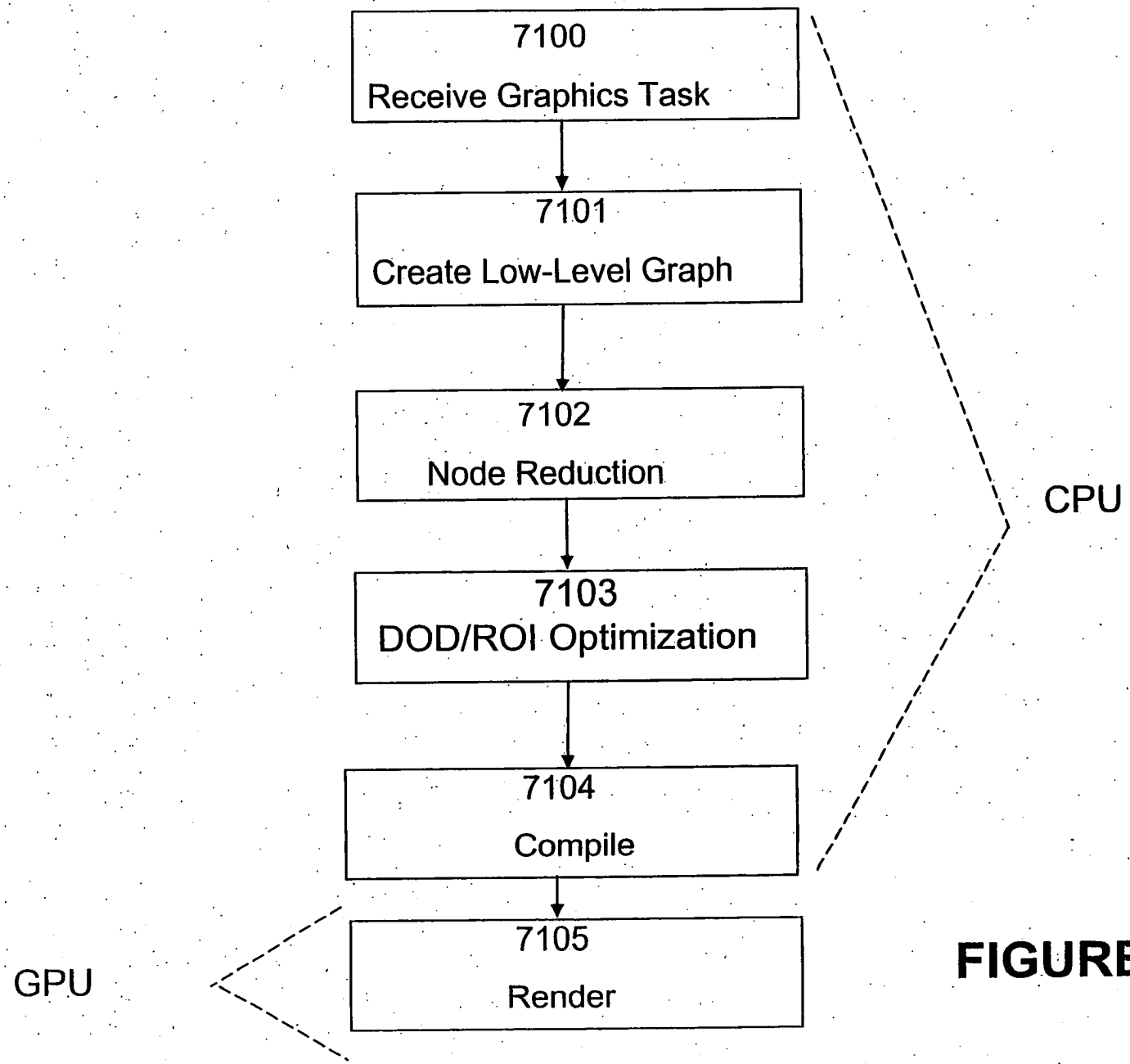

7100
Receive Graphics Task
7101
Create Low-Level Graph
7102
Node Reduction
7103
DOD/ROI Optimization
7104
Compile
7105
Render
CPU
GPU

FIGURE 7

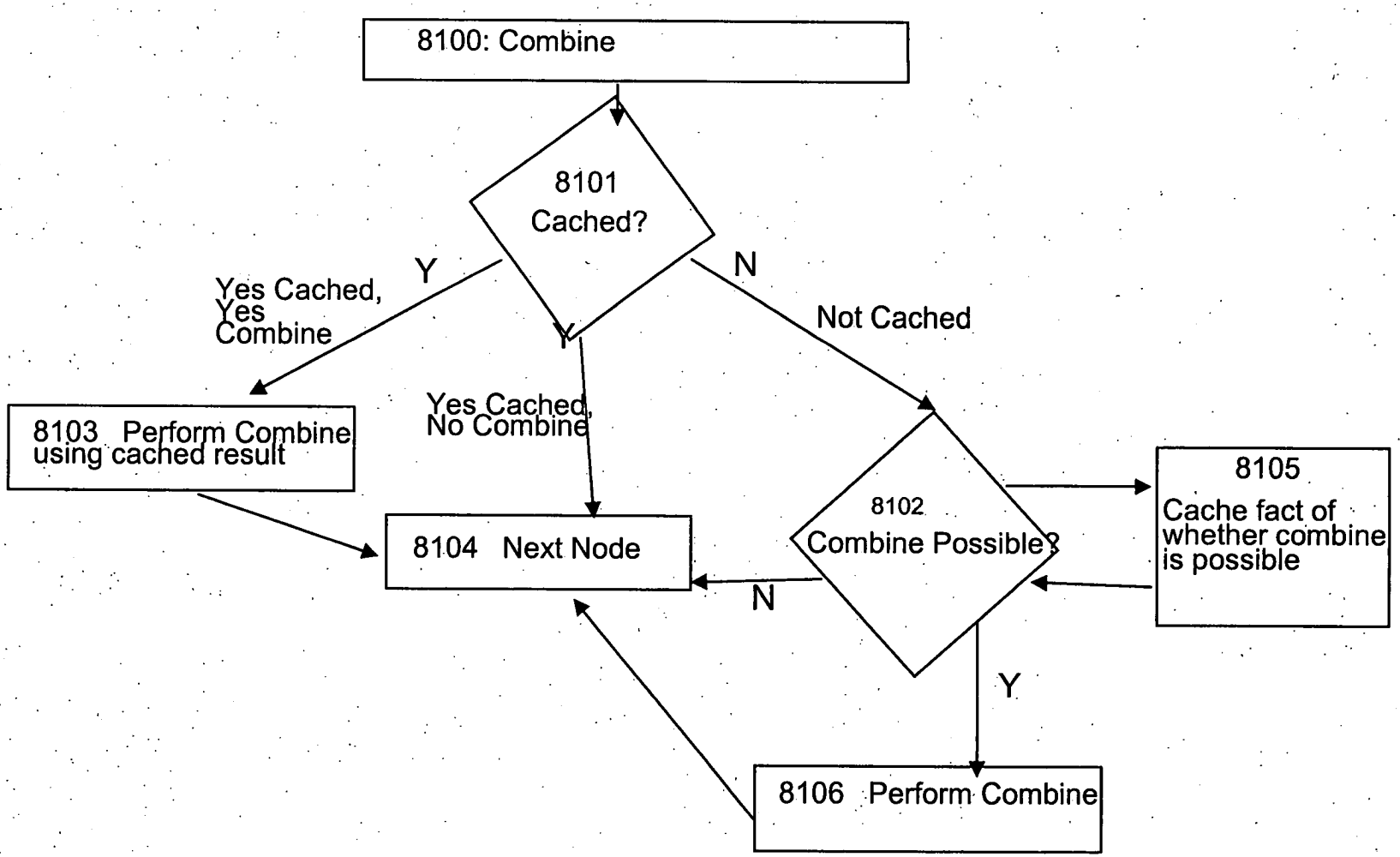

8100: Combine
8101
Cached?
Y
Yes Cached,
Yes
Combine
Yes Cached,
No Combine
N
Not Cached
8103 Perform Combine
using cached result
8104 Next Node
8102
Combine Possible?
N
8105
Cache fact of
whether combine
is possible
Y
8106 Perform Combine

FIGURE 8

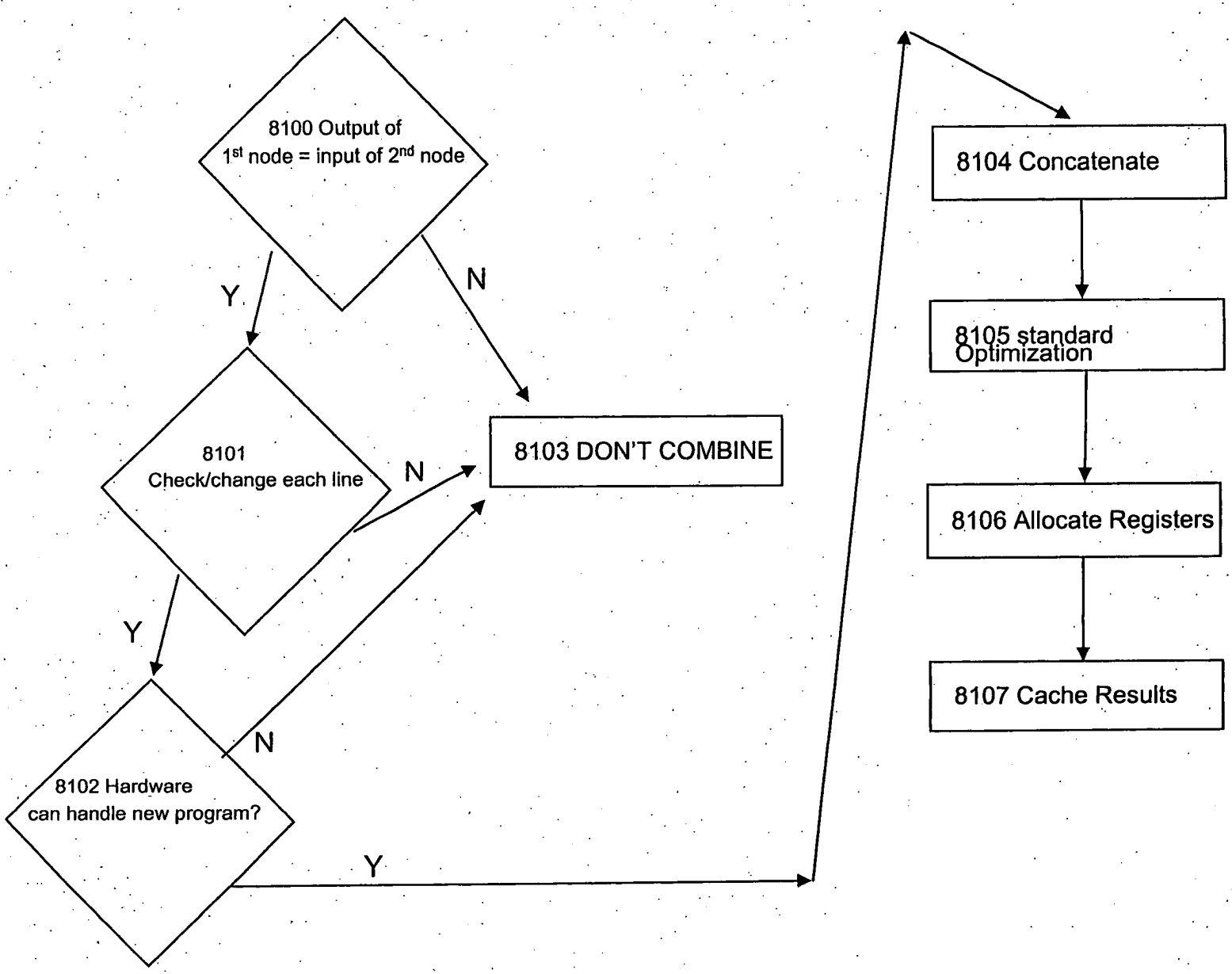

8100 Output of
1st node = input of 2nd node
Y
N
8101
Check/change each line
N
8103 DON'T COMBINE
Y
N
8102 Hardware
can handle new program?
Y
8104 Concatenate
8105 standard
Optimization
8106 Allocate Registers
8107 Cache Results

FIGURE 9

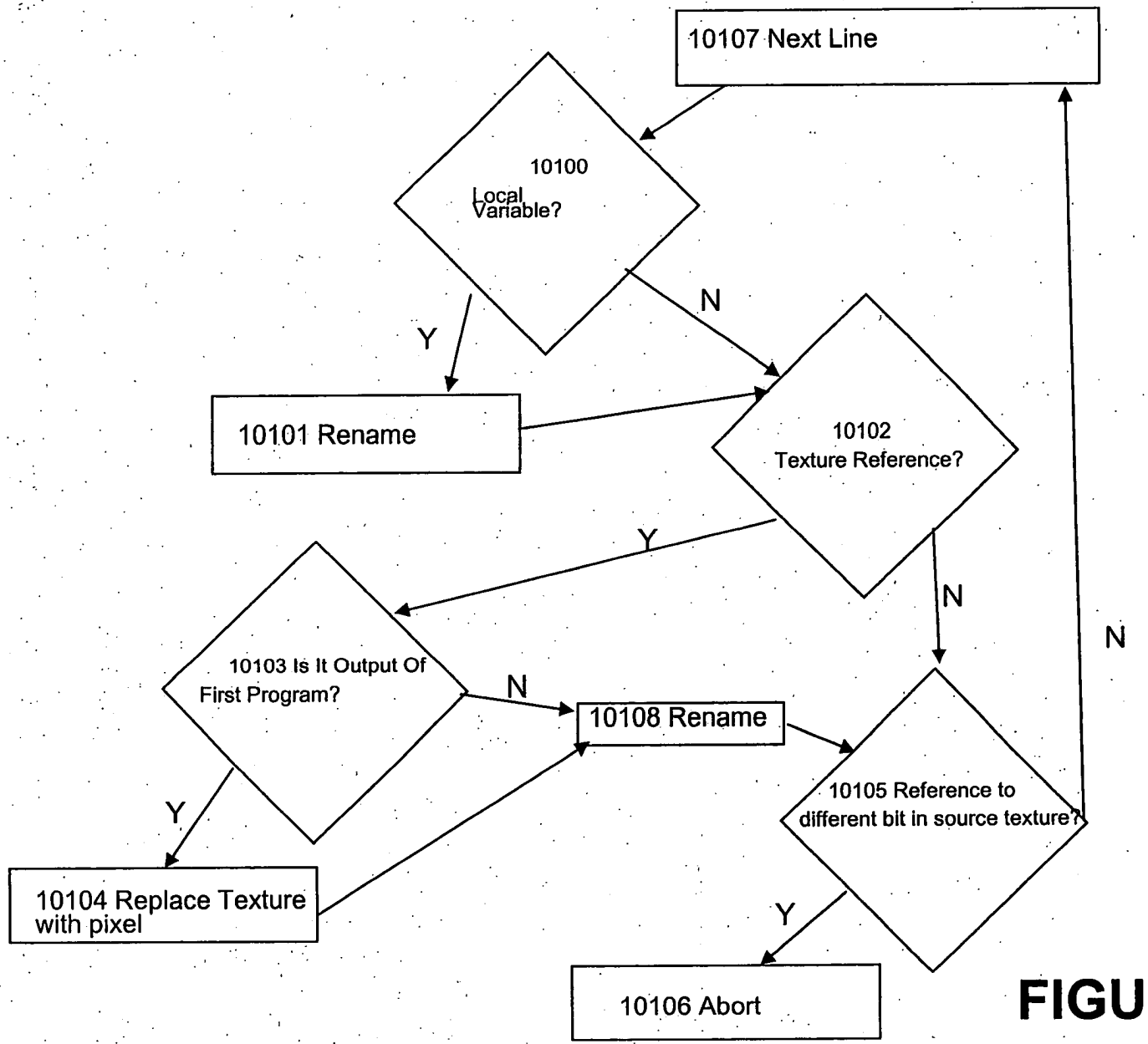
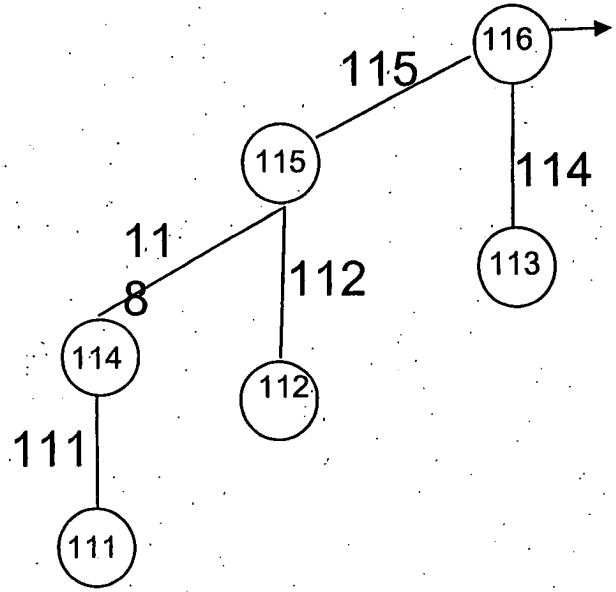
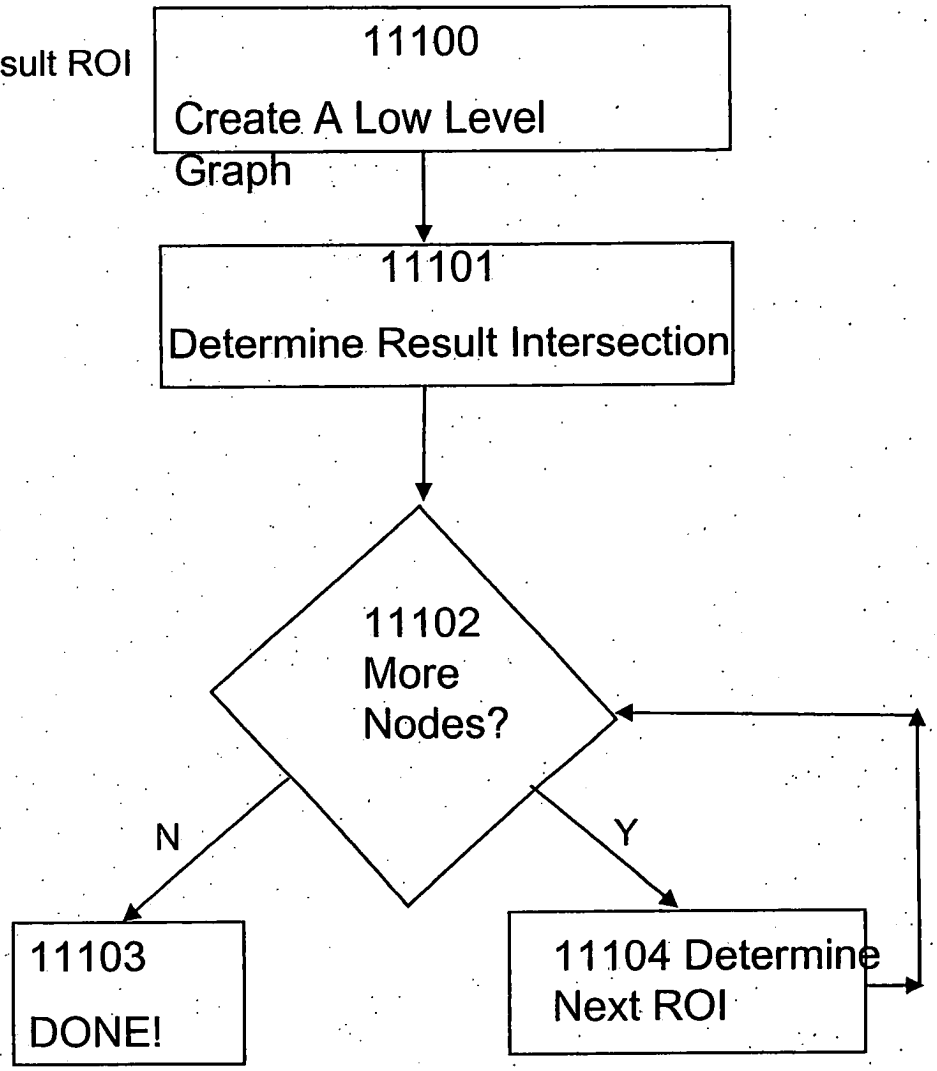

10107 Next Line
10100
Local Variable?
Y
N
10101 Rename
10102
Texture Reference?
Y
N
N
10103 Is It Output Of First Program?
N
10108 Rename
10105 Reference to different bit in source texture?
Y
10104 Replace Texture with pixel
Y
10106 Abort

FIGURE 10



11(a)



11(b)

Figure 11

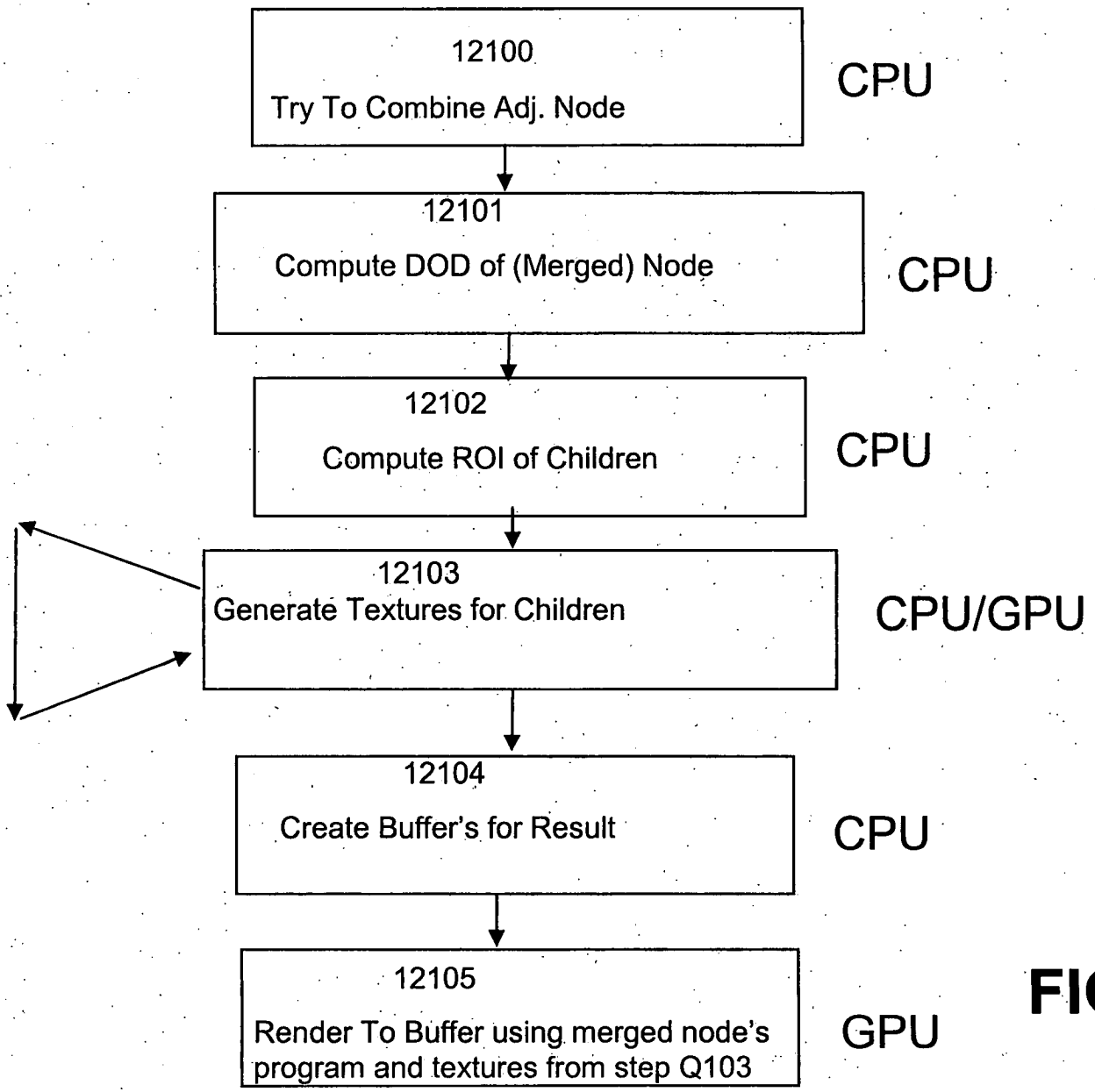

12100
Try To Combine Adj. Node
CPU
12101
Compute DOD of (Merged) Node
CPU
12102
Compute ROI of Children
CPU
12103
Generate Textures for Children
CPU/GPU
12104
Create Buffer's for Result
CPU
12105
Render To Buffer using merged node's program and textures from step Q103
GPU

FIGURE 12

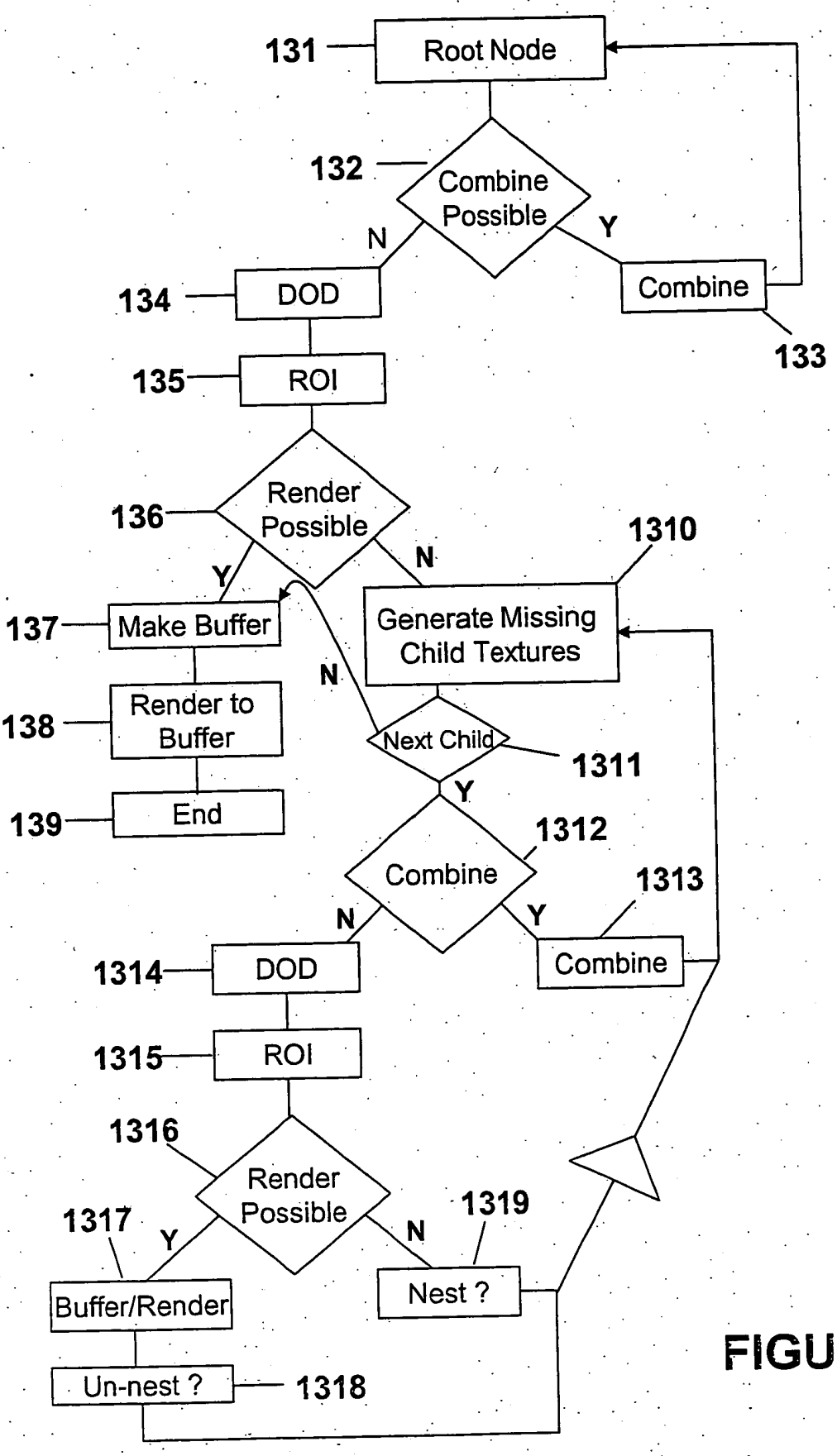

131
Root Node
132
Combine Possible
N
Y
134
DOD
Combine
133
135
ROI
136
Render Possible
Y
N
1310
137
Make Buffer
Generate Missing Child Textures
N
138
Render to Buffer
Next Child
1311
139
End
Y
1312
Combine
1313
N
Y
1314
DOD
Combine
1315
ROI
1316
Render Possible
1317
Y
N
1319
Buffer/Render
Nest ?
Un-nest ?
1318

FIGURE 13

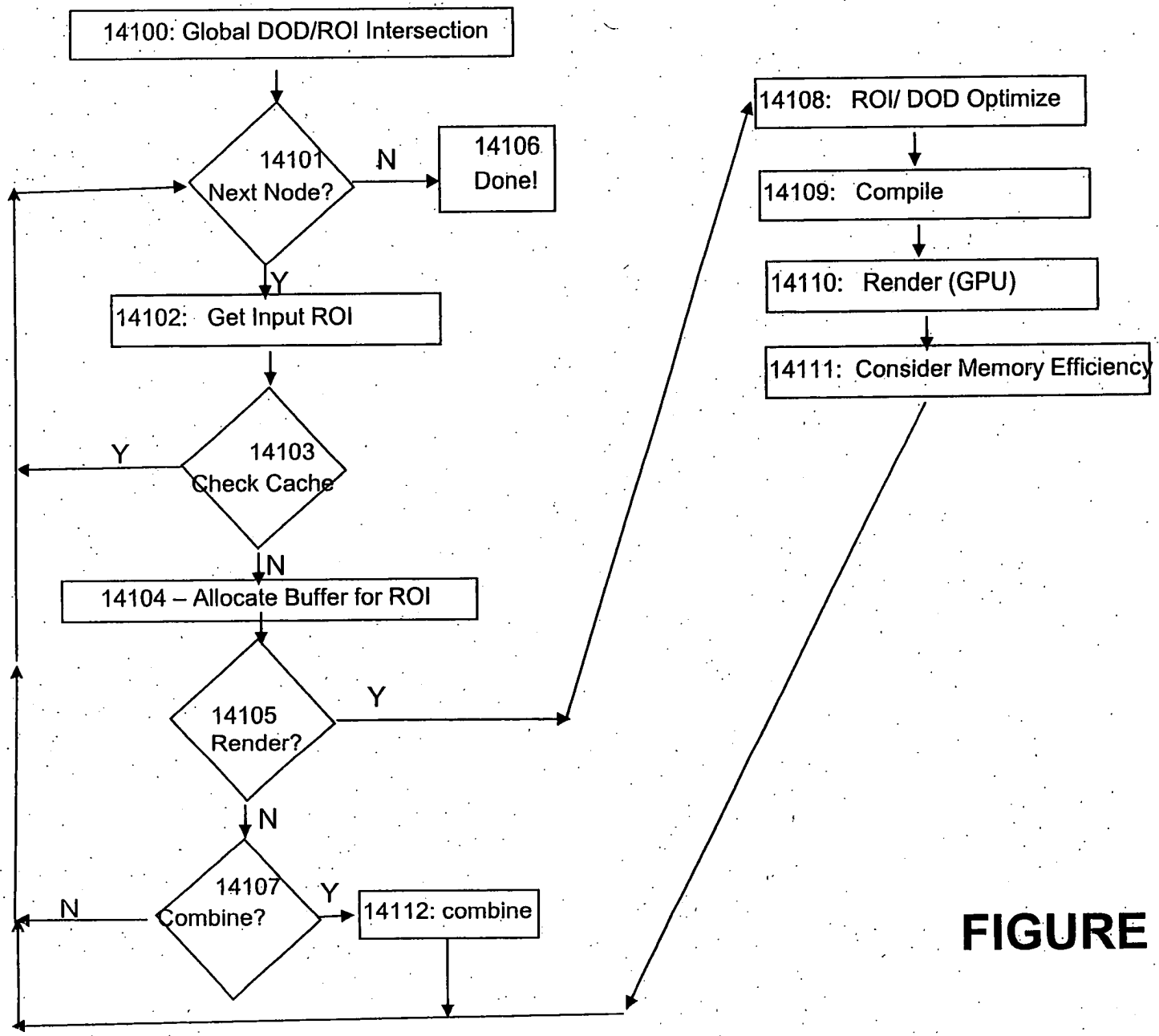

14100: Global DOD/ROI Intersection
14101 Next Node?
N
14106 Done!
Y
14102: Get Input ROI
14103 Check Cache
Y
N
14104 – Allocate Buffer for ROI
14105 Render?
Y
N
14107 Combine?
N
Y
14112: combine
14108: ROI/ DOD Optimize
14109: Compile
14110: Render (GPU)
14111: Consider Memory Efficiency

FIGURE 14

	CPU	GPU
Time 1	Frame 1	Prior Task
Time 2	Frame 2	Frame 1
Time 3	Frame 3	Frame 2
Time 4	Frame 4	Frame 3
Time 5	Frame 5	Frame 4
Time 6	Frame 6	Frame 5
Time 7	Frame 7	Frame 6

15(a)

FIGURE 15(a) & 15(b)

	P1	P2	P3	P4
Time 1	Frame 1			
Time 2	Frame 2	Frame 1		
Time 3	Frame 3	Frame 2	Frame 1	
Time 4	Frame 4	Frame 3	Frame 2	Frame 1
Time 5	Frame 5	Frame 4	Frame 3	Frame 2
Time 6	Frame 6	Frame 5	Frame 4	Frame 3
Time 7		Frame 6	Frame 5	Frame 4

15(b)

	CPU	GPU
Time 1	Frame 1, effect 1	
Time 2	Frame 2, effect 1	Frame 1, effect 2
Time 3	Frame 1, effect 3	Frame 2, effect 2
Time 4	Frame 2, effect 3	Frame 1, effect 4
Time 5	Frame 3, effect 1	Frame 2, effect 4
Time 6	Frame 4, effect 1	Frame 3, effect 2
Time 7	Frame 3, effect 3	Frame 4, effect 2
Time 8	Frame 4, effect 3	Frame 3, effect 4
		Frame 4, effect 4

FIGURE 15(c)

15(c)

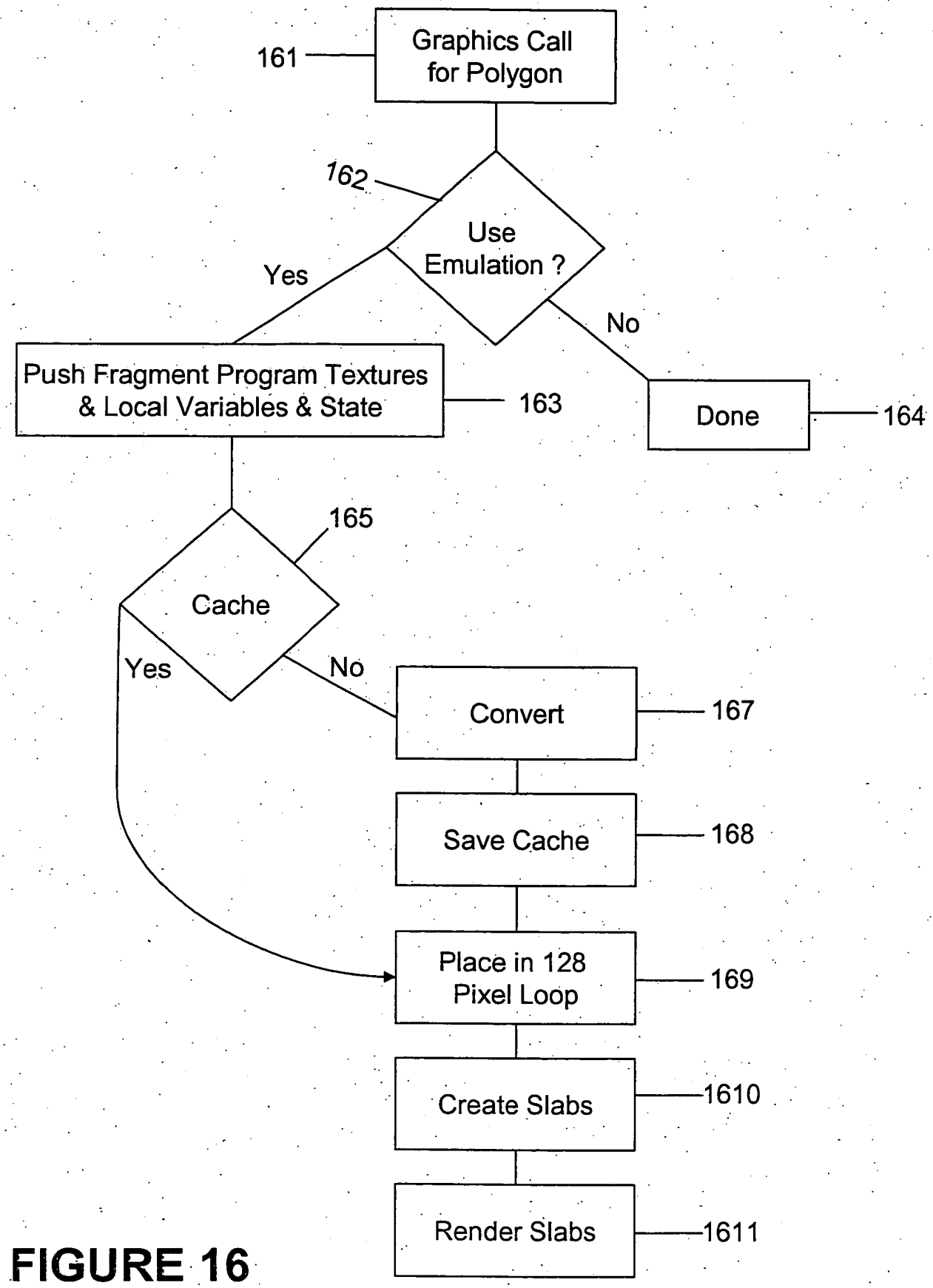
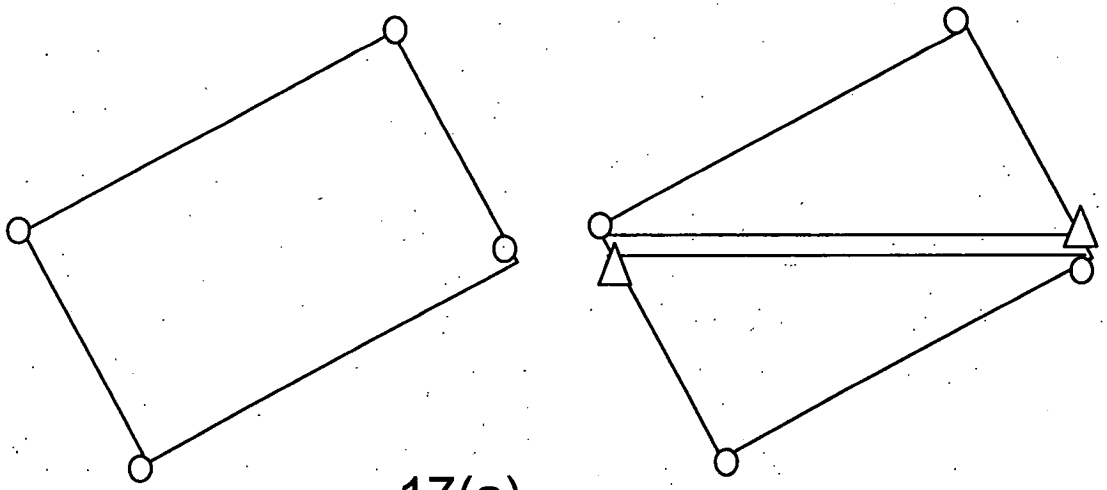
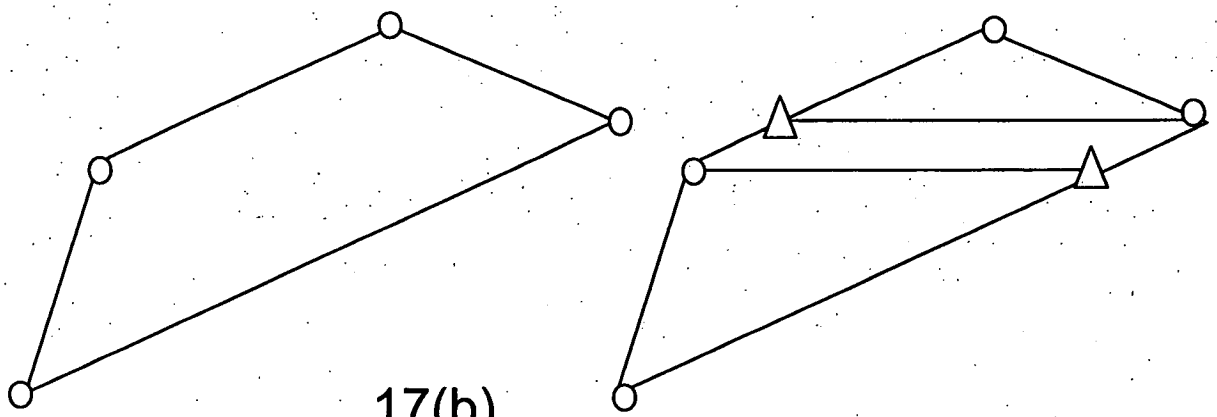

161
Graphics Call for Polygon
162
Use Emulation ?
Yes
No
Push Fragment Program Textures & Local Variables & State
163
Done
164
165
Cache
Yes
No
Convert
167
Save Cache
168
Place in 128 Pixel Loop
169
Create Slabs
1610
Render Slabs
1611

FIGURE 16



17(a)



17(b)

Figures 17(a) & 17(b)